

**In the Claims:**

Please amend claims 14, 18, 27 – 29 and 41, as indicated below.

1. (Original) A system, comprising:

a processor;

a computer-accessible medium coupled to the processor, wherein the computer-accessible medium is configured to store program instructions executable by the processor to implement an application program comprising:

one or more application modules, wherein at least a first one of the application modules comprises a first dynamic component and a static component, wherein the first dynamic component and the static component are configured to function according to an initial set of requirements for the application; and

a dynamic component generator configured to receive a new set of requirements for the application and generate a second dynamic component to replace the first dynamic component, wherein the second dynamic component is configured to function according to the new set of requirements.

2. (Original) The system as recited in claim 1, wherein the dynamic component generator does not change the static component in response to the new set of requirements.

3. (Original) The system as recited in claim 1, wherein the dynamic component generator is configured to generate a second dynamic component to replace the first dynamic component by modifying the first dynamic component in response to the new

set of requirements.

4. (Original) The system as recited in claim 1, wherein the dynamic component generator is configured to replace the first dynamic component by overwriting the first dynamic component in the computer-accessible medium in response to the new set of requirements.

5. (Original) The system as recited in claim 1, wherein the new set of requirements is formatted according to an eXtensible Mark-up Language (XML) schema and stored in the computer-accessible medium.

6. (Original) The system as recited in claim 1, wherein the one or more application modules comprise a second application module comprising a static component and a dynamic component.

7. (Original) The system as recited in claim 6, wherein the dynamic component generator is configured to generate a new dynamic component for the second application module in response to receiving the new set of requirements.

8. (Original) The system as recited in claim 6, further comprising another dynamic component generator for the dynamic component of the second application module, wherein the other dynamic component generator is configured to generate a new dynamic component for the second application module in response to receiving a new set of requirements for the second application module.

9. (Original) The system as recited in claim 1, wherein the first application module is a controller module, wherein the static component is a router component configured to receive user input, and wherein the dynamic component is an application logic component coupled to the router component, wherein the application logic component is configured to function according to a current set of application requirements in response to the user input.

10. (Original) The system as recited in claim 9, wherein the application logic component comprises an Enterprise Java Bean (EJB) session bean.

11. (Original) The system as recited in claim 1, wherein the first application module is a model module, wherein the static component is a static data model configured to function independent of an application data representation, and wherein the dynamic component is a dynamic data model configured to function dependent upon the application data representation and according to a current set of application requirements in response to the user input.

12. (Original) The system as recited in claim 11, wherein the dynamic data model comprises an Enterprise Java Bean (EJB) entity bean.

13. (Original) The system as recited in claim 11, wherein the new set of requirements indicates a change to the application data representation, and wherein the dynamic component generator is configured to generate a new dynamic data model in response to the change to the application data representation.

14. (Currently amended) A method, comprising:

installing one or more application modules each comprising a static component;

one or more dynamic component generators receiving an initial set of requirements for the application modules; and

the one or more dynamic component generators generating one or more initial dynamic components for the one or more application modules, wherein the one or more initial dynamic components are configured to function according to the initial set of requirements.

15. (Original) The method as recited in claim 14, further comprising:

receiving a new set of requirements for the application modules; and

generating one or more new dynamic components to replace the one or more initial dynamic components, wherein the one or more new dynamic components are configured to function according to the new set of requirements.

16. (Original) The method as recited in claim 15, wherein said generating one or more new dynamic components comprises replacing the one or more initial dynamic components by the one or more new dynamic components by modifying the each of the one or more initial dynamic components in response to the new set of requirements.

17. (Original) The method as recited in claim 15, wherein said generating one or more new dynamic components comprises replacing the one or more initial dynamic components by the one or more new dynamic components by overwriting each of the one or more initial dynamic components in a computer-accessible medium in response to the new set of requirements.

18. (Currently amended) The method as recited in claim 14, ~~wherein said generating is performed by one or more dynamic component generators~~, wherein the one or more dynamic component generators are comprised within the same application as the one or more application modules.

19. (Original) The method as recited in claim 14, wherein said generating is performed by one or more dynamic component generators comprised within an application server container, wherein the application modules are comprised within the same application server container.

20. (Original) The method as recited in claim 14, wherein, in said generating, the

static components comprised by the one or more application modules are not changed in response to the new set of requirements.

21. (Original) The method as recited in claim 14, wherein the new set of requirements is formatted according to an eXtensible Mark-up Language (XML) schema.

22. (Original) The method as recited in claim 14, wherein one of the one or more application modules is a controller module, wherein the static component is a router component configured to receive user input, and wherein a dynamic component generated for the one of the one or more application modules is an application logic component coupled to the router component, wherein the application logic component is configured to function according to a current set of requirements in response to the user input.

23. (Original) The method as recited in claim 22, wherein the application logic component comprises an Enterprise Java Bean (EJB) session bean.

24. (Original) The method as recited in claim 14, wherein one of the one or more application modules is a model module, wherein the static component is a static data model configured to function independent of an application data representation, and wherein a dynamic component generated for the one of the one or more application modules is a dynamic data model configured to function dependent upon the application data representation and according to a current set of requirements in response to the user input.

25. (Original) The method as recited in claim 24, wherein the dynamic data model comprises an Enterprise Java Bean (EJB) entity bean.

26. (Original) The method as recited in claim 24, further comprising:

receiving a new set of requirements indicating a change to the application data

representation; and

generating a new dynamic data model in response to the change to the application data representation.

27. (Currently amended) A method, comprising:

installing one or more application modules, wherein at least a first one of the application modules comprises a first dynamic component and a static component, wherein the first dynamic component and the static component are configured to function according to an initial set of requirements for the application;

one or more dynamic component generators receiving a new set of requirements for the application modules; and

the one or more dynamic component generators generating a new dynamic component to replace the first dynamic component, wherein the new dynamic component is configured to function according to the new set of requirements.

28. (Currently amended) The method as recited in claim 27, ~~wherein said generating is performed by one or more dynamic component generators~~, wherein the one or more dynamic component generators are comprised within the same application server as the one or more application modules.

29. (Currently amended) The method as recited in claim 27, wherein the said generating is performed by one or more dynamic component generator are comprised within an application server container, wherein the one or more application modules are comprised within the same application server container.

30. (Original) The method as recited in claim 27, wherein, in said generating, the static component does not change in response to the new set of requirements.

31. (Original) The method as recited in claim 27, wherein, in said generating, the second dynamic component replaces the first dynamic component by modifying the first dynamic component in response to the new set of requirements.

32. (Original) The method as recited in claim 27, wherein, in said generating, the second dynamic component replaces the first dynamic component by overwriting the first dynamic component in a computer-accessible medium in response to the new set of requirements.

33. (Original) The method as recited in claim 27, wherein the new set of requirements is formatted according to an eXtensible Mark-up Language (XML) schema and stored in the computer-accessible medium.

34. (Original) The method as recited in claim 27, wherein the one or more application modules comprise a second application module comprising a static component and a dynamic component.

35. (Original) The method as recited in claim 34, further comprising generating a new dynamic component for the second application module in response to receiving the new set of requirements.

36. (Original) The method as recited in claim 27, wherein the first application module is a controller module, wherein the static component is a router component configured to receive user input, and wherein the dynamic component is an application logic component coupled to the router component, wherein the application logic component is configured to function according to a current set of application requirements in response to the user input.

37. (Original) The method as recited in claim 36, wherein the application logic component comprises an Enterprise Java Bean (EJB) session bean.

38. (Original) The method as recited in claim 27, wherein the first application module is a model module, wherein the static component is a static data model configured to function independent of an application data representation, and wherein the dynamic component is a dynamic data model configured to function dependent upon the application data representation and according to a current set of application requirements in response to the user input.

39. (Original) The method as recited in claim 38, wherein the dynamic data model comprises an Enterprise Java Bean (EJB) entity bean.

40. (Original) The method as recited in claim 38, wherein the new set of requirements indicates a change to the application data representation, and wherein, in said generating, a new dynamic data model is generated in response to the change to the application data representation.

41. (Currently amended) A tangible computer accessible medium comprising program instructions, wherein the program instructions are executable by a processor to implement an application program comprising:

one or more application modules, wherein at least a first one of the application modules comprises a first dynamic component and a static component, wherein the first dynamic component and the static component are configured to function according to an initial set of requirements for the application; and

a dynamic component generator configured to receive a new set of requirements for the application and generate a second dynamic component to replace



the first dynamic component, wherein the second dynamic component is configured to function according to the new set of requirements.

42. (Original) The medium as recited in claim 41, wherein the dynamic component generator does not change the static component in response to the new set of requirements.

43. (Original) The medium as recited in claim 41, wherein the dynamic component generator is configured to generate a second dynamic component to replace the first dynamic component by modifying the first dynamic component in response to the new set of requirements.

44. (Original) The medium as recited in claim 41, wherein the dynamic component generator is configured to replace the first dynamic component by overwriting the first dynamic component in the computer-accessible medium in response to the new set of requirements.

45. (Original) The medium as recited in claim 41, wherein the new set of requirements is formatted according to an eXtensible Mark-up Language (XML) schema and stored in the computer-accessible medium.

46. (Original) The medium as recited in claim 41, wherein the one or more application modules comprise a second application module comprising a static component and a dynamic component.

47. (Original) The medium as recited in claim 46, wherein the dynamic component generator is configured to generate a new dynamic component for the second application module in response to receiving the new set of requirements.

48. (Original) The medium as recited in claim 46, further comprising another dynamic component generator for the dynamic component of the second application

module, wherein the other dynamic component generator is configured to generate a new dynamic component for the second application module in response to receiving a new set of requirements for the second application module.

49. (Original) The medium as recited in claim 41, wherein the first application module is a controller module, wherein the static component is a router component configured to receive user input, and wherein the dynamic component is an application logic component coupled to the router component, wherein the application logic component is configured to function according to a current set of application requirements in response to the user input.

50. (Original) The medium as recited in claim 49, wherein the application logic component comprises an Enterprise Java Bean (EJB) session bean.

51. (Original) The medium as recited in claim 41, wherein the first application module is a model module, wherein the static component is a static data model configured to function independent of an application data representation, and wherein the dynamic component is a dynamic data model configured to function dependent upon the application data representation and according to a current set of application requirements in response to the user input.

52. (Original) The medium as recited in claim 51, wherein the dynamic data model comprises an Enterprise Java Bean (EJB) entity bean.

53. (Original) The medium as recited in claim 51, wherein the new set of requirements indicates a change to the application data representation, and wherein the dynamic component generator is configured to generate a new dynamic data model in response to the change to the application data representation.